ANALYSIS OF WATERMARKING SYSTEMS IN THE FREQUENCY DOMAIN AND ITS APPLICATION TO DESIGN OF ROBUST WATERMARKING SYSTEMS

Akio MIYAZAKI and Akihiro OKAMOTO

Graduate School of Information Science and Electrical Engineering, Kyushu University, Fukuoka, JAPAN

ABSTRACT

This paper aims to design robust watermarking systems in the frequency domain. We first present the general model of watermark embedding and extracting processes and carry out their analysis. Then we examine the robustness of the watermarking system against common image processing and clarify the reason why detection errors occur in the watermark extracting process. Based on the result, we improve the watermark extracting process and design the robust watermarking systems. The improvement is accomplished using deconvolution filter and neural network techniques. Numerical experiments using the DCT-based watermarking system show good performance as we expected.

1. INTRODUCTION

With the rapid spread of computer networks and the further development of multimedia technologies, the copyright protection of digital contents such as audio, image and video, has been one of the most serious problems because digital copies can be made identical to the original. The digital watermark technology is now drawing the attention as a new method of protecting copyrights of digital contents. Digital watermark is realized by embedding information data directly into digital contents with an imperceptible form for human audio/visual systems, and should satisfy the following requirements: The embedded watermark does not spoil the quality of the original contents and should not be perceptilbe. It must be difficult for an attacker to remove the watermark and should be robust to common signal processing and geometric distortions.

There are mainly two methods of the digital watermark technology for still images. One is embedding in the spatial domain. The other is embedding in the frequency domain. It is generally said that embedding in the frequency domain is more tolerant of attacks and image processing than in the spatial domain. Thus, most of recently proposed methods[1] embed the watermark into the spectral coefficients of images by using the signal transformation such as the discrete cosine transformation (DCT) or the discrete wavelet transformation (DWT).

However, in these methods, no theoretical limits to their robustness against attacks and image processing have not been given, and the watermarking methods that are robust to almost common image processing have not appear yet. On the contrary, the watermark-removal softwares, such as StirMark[2] and unZign[3], heve succeeded in washing the watermark away for most of watermarking systems. Such a situation is quite discouraging but will foster new research in this field such as the analysis of the performance of watermarking systems and the development of watermarking systems with the desired robustness. Therefore, as the first stage in the development of watermarking technology, it is important to analyze the watermark embedding and extracting processes and apply the result to the design of robust watermarking systems.

In this paper, we concentrate on the watermarking of still images. We first present the general model of a watermarking method in the frequency domain and analyze the watermarking system. Then we investigate the robustness of the watermarking system against common image processing. It is clarified how distortion occurs to the watermark by image processing and the reason why the distortion causes detection errors in the watermark extracting procedure. Based on the result, we further carry out an improvement of the watermark extracting process by designing a deconvolution filter that attempts to undo the effect of watermark distortion. The design is achieved using the neural network technique. Numerical experiments with the DCT-based watermarking system show good performance as we wished.

2. ANALYSIS OF WATERMARKING SYSTEMS IN THE FREQUENCY DOMAIN

We analyze the watermark embedding and extracting processes of a watermarking method in the frequency domain.

In the watermark embedding process, an (original) $M \times M$ image $\boldsymbol{s} = [s(m,n)]$, where $s(m,n)$ denotes a pixel quantized to 256 levels (represented by 8 bits), is first converted into a spectral coefficient $\boldsymbol{c} = [c(m,n)]$ by using the signal transformation such as the DCT or DWT as

$$c(m,n) = \sum_{k=0}^{M-1} \sum_{l=0}^{M-1} \phi(m,n;k,l) s(k,l), \tag{1}$$

$\phi(m,n;k,l)$ being the 2D transfrom kernel. For simplicity of description, putting $N = M^2$, we now map the $M \times M$ image array $\boldsymbol{s} = [s(m,n)]$ and the $M \times M$ coefficient array $\boldsymbol{c} = [c(m,n)]$ into vectors $\boldsymbol{s} = [s(n)]$ and $\boldsymbol{c} = [c(n)]$ of size N, respectively, each by row ordering, and rewrite the 2D transform of size M in Eq. (1) as a 1D transform of size N

$$\boldsymbol{c} = \boldsymbol{T}\boldsymbol{s}, \tag{2}$$

where $\boldsymbol{T}$ is an $N \times N$ matrix created from the 2D transfrom kernel $\phi(m,n;k,l)$.

Next, we select B spectral coefficients $c(i_1)$, $c(i_2)$, $\cdots, c(i_B)$ from the $\boldsymbol{c}$ and quantize $c(i_k)$ to $c_Q(i_k)$ $(1 \leq k \leq B)$. Then, a watermark $w(k)$ $(1 \leq k \leq B)$, which is a binary data, *i.e.,* $w(k) = 1$ or -1, is embedded into $c_Q(i_k)$ in the form of

$$c'(i_k) = \begin{cases} c_Q(i_k) + \Delta, & w(k) = \ \ 1 \\ c_Q(i_k) - \Delta, & w(k) = -1 \end{cases} \tag{3}$$

where Δ is the embedded intensity, and the watermarked coefficient $\boldsymbol{d} = [d(n)]$ is made of

$$d(n) = \begin{cases} c'(i_k) & , \ n = i_k \ (1 \leq k \leq B) \\ c(n) & , \ \text{otherwise .} \end{cases} \tag{4}$$

It is noted that how to quantize the $c(i_k)$ depends on watermark embedding methods. In the following, let us show the representation of Eq. (3) in the case of watermark embedding using a controlled quantization process of spectral coefficients. Then the $c(i_k)$ is modified (quantized) as follows: Let Q be a quantization step size. Then we have, in Eq. (3), $c_Q(i_k) = (4l+1)Q/2$ when $c(i_k) \in [2lQ, (2l+1)Q)$, $c_Q(i_k) = (4l-1)Q/2$ when $c(i_k) \in [(2l-1)Q, 2lQ)$, and $\Delta = Q/2$, that is, Eq. (3) can be written as

$$c'(i_k) = \begin{cases} (2l+1)Q & , \ w(k) = \ \ 1 \\ 2lQ & , \ w(k) = -1 \end{cases} \tag{5}$$

$$\text{if} \quad c(i_k) \in [2lQ, (2l+1)Q)$$

and

$$c'(i_k) = \begin{cases} 2lQ & , \ w(k) = \ \ 1 \\ (2l-1)Q & , \ w(k) = -1 \end{cases} \tag{6}$$

$$\text{if} \quad c(i_k) \in [(2l-1)Q, 2lQ)$$

By defining the $N \times B$ matrix $\boldsymbol{E} = [e(m,n)]$ where

$$e(m,n) = \begin{cases} 1 & , \ (m,n) = (i_k, k) \\ 0 & , \ \text{otherwise} \end{cases} \tag{7}$$

and $1 \leq k \leq B$, the $\boldsymbol{d}$ can be written as

$$\boldsymbol{d} = \boldsymbol{c}_\text{o} + \Delta \boldsymbol{E}\boldsymbol{w} \tag{8}$$

where $\boldsymbol{c}_\text{o} = [c_0(n)]$ is the N-dimensional vector whose elements are given by

$$c_0(n) = \begin{cases} c_Q(i_k) & , \ n = i_k \ (1 \leq k \leq B) \\ c(n) & , \ \text{otherwise} \end{cases} \tag{9}$$

and $\boldsymbol{w} = [w(k)]$ is a B-dimensional watermark vector.

By the inverse transform $\boldsymbol{T}^{-1}$ of the $\boldsymbol{d}$, the watermarked image $\boldsymbol{x} = [x(n)]$ is obtained as

$$\boldsymbol{x} = \boldsymbol{T}^{-1}\boldsymbol{d} \tag{10}$$

and the pixels $x(n)$ are quantized to 256 levels (8 bits). As the result, we have the (quantized) watermarked image $\boldsymbol{y} = [y(n)]$. It is noted that $\boldsymbol{y}$ is represented as

$$\boldsymbol{y} = \text{Quantization}[\boldsymbol{x}] = \boldsymbol{x} + \boldsymbol{\delta}, \tag{11}$$

where $\boldsymbol{\delta} = [\delta(n)]$ denotes the quantization error whose elements are independent and probability law has a uniform probability density over the interval $[-0.5, 0.5)$.

The set of parameters $K = \{\Delta, \boldsymbol{E}, \boldsymbol{c}_Q\}$, where we put $\boldsymbol{c}_Q = [c_Q(i_k)]$ (B-dimensional vector), which is used as key data in the watermark detection. It is necessary to decide the K so that images may not be degraded through the watermark embedding.

In the watermark extracting process, we transform the watermarked image $\boldsymbol{z} = [z(n)]$ into the spectral coefficient $\boldsymbol{d}' = [d'(n)]$ by the transformation $\boldsymbol{T}$ as

$$\boldsymbol{d}' = \boldsymbol{T}\boldsymbol{z}, \tag{12}$$

and pick up the watermarked coefficients $\boldsymbol{u}$ by

$$\boldsymbol{u} = \boldsymbol{E}_t \boldsymbol{d}', \tag{13}$$

where $\boldsymbol{E}_t$ denotes the transpose of $\boldsymbol{E}$ and satisfies $\boldsymbol{E}_t\boldsymbol{E} = \boldsymbol{I}$ (the $B \times B$ unit matrix). When $\boldsymbol{z} = \boldsymbol{y}$, we have, from Eqs. (8), (10) and (11),

$$\boldsymbol{d}' = \boldsymbol{c}_\text{o} + \Delta \boldsymbol{E}\boldsymbol{w} + \boldsymbol{T}\boldsymbol{\delta}. \tag{14}$$

Considering Eqs. (13), (14) and $\boldsymbol{E}_t\boldsymbol{c}_\text{o} = \boldsymbol{c}_Q$, we get the following watermark extracting procedure with the key data $K = \{\Delta, \boldsymbol{E}, \boldsymbol{c}_Q\}$: Let $\boldsymbol{d}'$ be the spectral coefficient of a watermarked image $\boldsymbol{z}$. Then,

$$\left.\begin{array}{ll} \text{(i)} & \boldsymbol{u} = \boldsymbol{E}_t \boldsymbol{d}' \\ \text{(ii)} & \boldsymbol{v} = \boldsymbol{u} - \boldsymbol{c}_Q \\ \text{(iii)} & \boldsymbol{w}' = g(\boldsymbol{v}) \end{array}\right\} \tag{15}$$

where $g(\cdot)$ is the step function $g(v) = 1$ for $v \geq 0$; $= -1$ for $v < 0$. In order that the watermark is extracted correctly by using the watermark detector (15), that is, we have $\boldsymbol{w}' = \boldsymbol{w}$, we can see that the K should be set so as to satisfy the condition $\mid \varepsilon_0(k) \mid < \Delta$, where $\boldsymbol{\varepsilon}_0 = [\varepsilon_0(k)] = \boldsymbol{E}_t\boldsymbol{T}\boldsymbol{\delta}$, because $\boldsymbol{v} = \Delta\boldsymbol{w} + \boldsymbol{\varepsilon}_0$ when $\boldsymbol{z} = \boldsymbol{y}$.

3. ROBUST WATERMARKING SYSTEMS

We consider the robustness of the watermarking system against common image processing. Let $\boldsymbol{f}$ be an image operator that represents a certain image processing, and let

$$\boldsymbol{z} = \boldsymbol{f}(\boldsymbol{y}) = [\ f_1(\boldsymbol{y}),\ f_2(\boldsymbol{y}),\ \cdots,\ f_N(\boldsymbol{y})\ ]_t, \qquad (16)$$

where $\boldsymbol{y}$ is a watermarked image. Then, from Eq. (8), (10) and (11), putting $\boldsymbol{s}_0 = \boldsymbol{T}^{-1}\boldsymbol{c}_0$, we have

$$\boldsymbol{y} \ = \ \boldsymbol{s}_0 + \Delta\boldsymbol{T}^{-1}\boldsymbol{E}\boldsymbol{w} + \boldsymbol{\delta}. \qquad (17)$$

Since watermarked images are not degraded through the watermark embedding, *i.e.*, $\|\boldsymbol{s}_0\| \gg \|\ \Delta\boldsymbol{T}^{-1}\boldsymbol{E}\boldsymbol{w} + \boldsymbol{\delta}\ \|$, $\|\cdot\|$ being the norm of vectors,

$$\begin{aligned} \boldsymbol{z} &= \boldsymbol{f}(\boldsymbol{s}_0 + \Delta\boldsymbol{T}^{-1}\boldsymbol{E}\boldsymbol{w} + \boldsymbol{\delta}) \\ &\simeq \boldsymbol{f}(\boldsymbol{s}_0) + \boldsymbol{F}(\boldsymbol{s}_0)(\Delta\boldsymbol{T}^{-1}\boldsymbol{E}\boldsymbol{w} + \boldsymbol{\delta}) \qquad (18) \end{aligned}$$

is obtained, where $f_m = f_m(\boldsymbol{y})$, $y_n = y(n)$ and

$$\boldsymbol{F}(\boldsymbol{s}_0) = [f_{m,n}(\boldsymbol{s}_0)] = \left[\left. \frac{\partial f_m}{\partial y_n} \right|_{\boldsymbol{y}=\boldsymbol{s}_0} \right]. \qquad (19)$$

Hence, from Eqs. (12) and (13), distortion occurs to the watermarked coefficient $\boldsymbol{u}$ as

$$\begin{aligned} \boldsymbol{u} &= \boldsymbol{E}_t\boldsymbol{T}\boldsymbol{f}(\boldsymbol{s}_0) + \Delta\boldsymbol{E}_t\boldsymbol{T}\boldsymbol{F}(\boldsymbol{s}_0)\boldsymbol{T}^{-1}\boldsymbol{E}\boldsymbol{w} \\ &+ \boldsymbol{E}_t\boldsymbol{T}\boldsymbol{F}(\boldsymbol{s}_0)\boldsymbol{\delta}, \qquad (20) \end{aligned}$$

and detection errors arise in the watermark extracting procedure.

[Remark] It is noted that in the case of linear transformation, $\boldsymbol{z} = \boldsymbol{F}\boldsymbol{y}$, Eq. (20) can be written as

$$\begin{aligned} \boldsymbol{u} &= \boldsymbol{E}_t\boldsymbol{T}\boldsymbol{F}\boldsymbol{s}_0 + \Delta\boldsymbol{E}_t\boldsymbol{T}\boldsymbol{F}\boldsymbol{T}^{-1}\boldsymbol{E}\boldsymbol{w} \\ &+ \boldsymbol{E}_t\boldsymbol{T}\boldsymbol{F}\boldsymbol{\delta}. \qquad (21) \end{aligned}$$

We can see from Eq. (20) that the watermark $\boldsymbol{w}$ is convolved with the filter, incorporating the effect of distortion, whose impulse response is described by the elements of the $B \times B$ matrix $\boldsymbol{E}_t\boldsymbol{T}\boldsymbol{F}(\boldsymbol{s}_0)\boldsymbol{T}^{-1}\boldsymbol{E}$, and further the bias $\boldsymbol{E}_t\boldsymbol{T}\boldsymbol{f}(\boldsymbol{s}_0)$ and the noise $\boldsymbol{E}_t\boldsymbol{T}\boldsymbol{F}(\boldsymbol{s}_0)\boldsymbol{\delta}$ corrupt the watermark $\boldsymbol{w}$. This implies that we can design a deconvolution filter that attempts to undo the effects of the convolution filter and the bias as follows: From Eq. (20), putting $\boldsymbol{H} = (\boldsymbol{E}_t\boldsymbol{T}(\boldsymbol{F}(\boldsymbol{s}_0)\boldsymbol{T}^{-1}\boldsymbol{E})^{-1}$ and $\boldsymbol{r} = -\boldsymbol{H}\boldsymbol{E}_t\boldsymbol{T}\boldsymbol{f}(\boldsymbol{s}_0)$, we have the improved watermark extracting procedure as

$$\left.\begin{array}{ll} \text{(i)} & \boldsymbol{u} = \boldsymbol{E}_t\boldsymbol{d}' \\ \text{(ii)} & \boldsymbol{v} = \boldsymbol{H}\boldsymbol{u} + \boldsymbol{r} \\ \text{(iii)} & \boldsymbol{w}' = g(\boldsymbol{v}) \end{array}\right\} \qquad (22)$$

In this watermark detector, we can get the correct estimate $w'(k)$ of the k-th watermark $w(k)$ provided that $\mid \varepsilon_1(k) \mid < \Delta$ where $\boldsymbol{\varepsilon}_1 = [\varepsilon_1(k)] = \boldsymbol{H}\boldsymbol{E}_t\boldsymbol{T}\boldsymbol{F}(\boldsymbol{s}_0)\boldsymbol{\delta}$, because from Eqs. (20) and (22), $\boldsymbol{v}$ is expressed as $\boldsymbol{v} = \Delta\boldsymbol{w} + \boldsymbol{\varepsilon}_1$.

4. DESIGN OF ROBUST WATERMARKING SYSTEMS USING NEURAL NETWORK

We consider that the robust watermarking system stated in Section 3 may be designed in application of the neural network technique because the deconvolution filter can be realized by a neural network. That is, by training the neural network, the network will produce automatically the adequate filter in accordance with the predescribed image operator $\boldsymbol{f}$. The design procedure is as follows:

We first construct the deconvolution filter (Eq. (22)) by a three-layered neural network

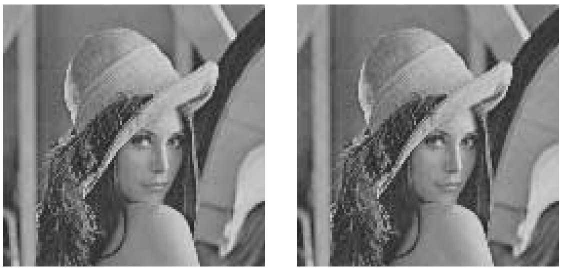
$$w'(k) = g\left(\sum_{l=1}^{B} h(k,l)u(l) + r(k) \right) \ (1 \leq k \leq B) \quad (23)$$

in which the function $g(v) = \tanh v$ is used instead of the step function. Then we train the neural network in order to undo the effects of the convolution filter and the bias in Eq. (20). The objective of the training process is that the weights $h(k,l)$'s and $r(k)$'s are set to the optimum values by minimizing the error $e(k) = w(k) - w'(k)$ between various training watermarks $\{w(k)\}$ and the outputs $\{w'(k)\}$ of the neural network. As is well known, one of the simple method for setting the weights to these values is the back-propagation algorithm[4]. Training take place during many trials or runs until the weights converge to the optimum values, that is, the neural network learns the characteristic of the deconvolution filter.

5. NUMERICAL EXPERIMENTS

In this section, we focus on the DCT-based watermarking method, in which the watermark is embedded into DCT coefficients of an image, properly selected, by using a controlled quantization process, and try to improve the watermark extracting process using the design technique described in Section 4.

In this experiment, we use the image LENNA with the size of 128 × 128 pixels (Figure 1 (a)) and the parameter $K = \{\Delta, \boldsymbol{E}, \boldsymbol{c}_Q\}$ where $\Delta = 20$, $B = 100$, and the watermark $w(k)$ is embedded into the (m, n) component ($30 \leq m, n \leq 39$) in the 128 × 128 DCT coefficient array. Figure 1 (b) shows the watermarked image.



(a) (b)

Figure 1 : (a) Original image s (LENNA). (b) The watermarked image y in which data of 100 bits are hidden. The root mean square (RMS) of $y - s$ is 0.98.

We first examine the robustness of the above watermarking system against smoothing with mean filter and luminance transformation with gamma correction. Figure 2 and 3 show the rate of the watermark detection error (bit error rate, BER) for 50 watermarked images. We can see from these figures that bit errors increase according as the watermarked image is degraded through these processings.

Next, we design the deconvolution filter by the neural network. Training data are 300 pairs of watermark $\{w(k)\}$ and watermarked coefficient $\{u(k)\}$, where watermarks used in training are different from 50 watermarks used in testing, and the initial value of weights is $h(k,l) = 1$ for $k = l$; $= 0$ for $k \neq l$ and $r(k) = c_Q(i_k)$. We illustrate the result of the watermark detection with the deconvolution filter in Figure 2 and 3, too. As the result, bit errors decrease in comparison with those in the watermarking system without the deconvolution filter. Thus, the watermark extracting process is improved as we expected.

6. CONCLUDING REMARKS

Using the proposed technique, we can design the robust DCT-based watermarking system against other image processing, *e.g.,* lossy compression such as JPEG, additive noise, reduction of grayscale level, and scaling. The design method can also be applied to other watermarking systems in the frequency domain such as the DWT-based watermarking system. We believe that based on the results we can develop and improve the watermarking technology. These results will be reported in forthcoming papers.

7. REFERENCES

[1] B. Macq (*Guest Editor*), Special Issue on Identification and Protection of Multimedia Information, Proc. of the IEEE, Vol.87, No.7, July 1999.

[2] StirMark: http://www.cl.cam.ac.uk/~fapp2/ watermarking/stirmark

[3] unZign: http://www.altern.com/watermark

[4] D. E. Rumelhart, et. al., PARALLEL DISTRIBUTED PROCESSING, Vols. I, II, MIT Press, 1986.

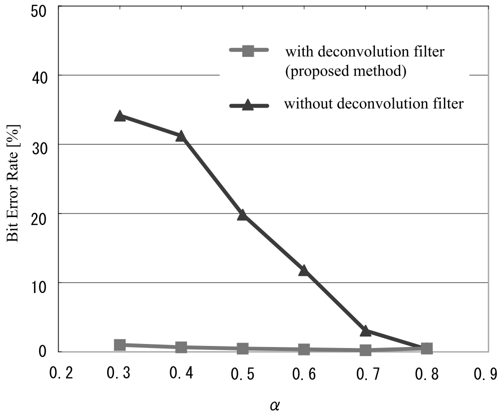


Figure 2 : Robustness of watermarking system against smoothing with mean filter:

$$z(m,n) = \sum_{k,l=-1}^{1} f(k,l)y(m-k, n-l)$$

where $f(k,l) = \alpha$ for $(k,l) = (0,0)$; $= (1-\alpha)/8$ for $(k,l) \neq (0,0)$ $(0 < \alpha < 1)$.

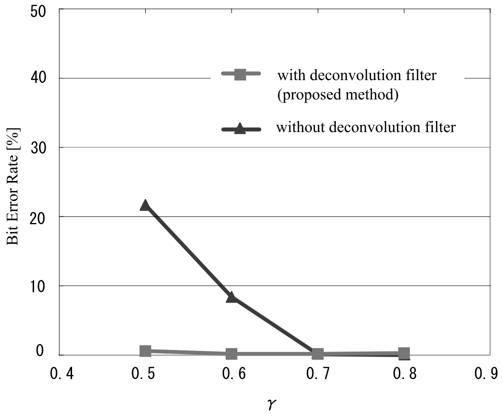


Figure 3 : Robustness of watermarking system against luminance transformation with gamma correction:

$$z(m,n) = 255^{1-\gamma} y(m,n)^{\gamma} \ (0 < \gamma < 1).$$